

## Claims

- [c1] 1. A search engine, comprising:
- a controller including a hash function able to receive an input search value and to create there from at least one hash address which is smaller in size than said input search value;
  - a hash pointer unit able to store a plurality of pointer values, wherein respective said pointer values are addressed based on said hash addresses;
  - a memory suitable for storing a database of search results, wherein respective said search results are addressed based on said pointer values;
  - an address bus operationally connecting said controller to said hash pointer unit and able to communicate said hash addresses from said controller to said hash pointer unit;
  - a pointer bus operationally connecting said hash pointer unit to said memory and able to communicate said pointer value from said hash pointer unit to said memory; and
  - a result bus operationally connecting said memory to said controller and able to communicate said search result from said memory to said controller, thereby permitting the search engine to function in a multi-way set-associative manner wherein the size of said memory is not a function of the degree of multi-way set-associativity.
- [c2] 2. The search engine of claim 1, wherein said pointer values are equal or smaller in size than said hash address.
- [c3] 3. The search engine of claim 1, wherein said search results each include a stored search value and an associate value, thereby permitting said controller to compare instances of said input search value with said stored search value to determine whether a hash collision has occurred and when said associate value is trustworthy.
- [c4] 4. The search engine of claim 3, wherein said hash function generates a hash value and said controller creates said hash address based on said hash value and an offset value.

[c7] 7. The search engine of claim 5, wherein said search results each include a stored search value and an associate value, thereby permitting said controller means to compare instances of said input search value with said stored search value to determine whether a hash collision has occurred and when said associate value is trustworthy.

Page 33 of 54

[c9] 9. A method for searching a database of search results, wherein the search results each include a stored search value, the method comprising the steps of:

- (a) generating a hash value from an input search value, wherein said hash value is smaller in size than said input search value;
- (b) creating a plurality of hash addresses based on said hash value and respective offset values;
- (c) retrieving pointer values from a pre-stored plurality of said pointer values based on said hash addresses;
- (d) retrieving instances of the search results from the database based on said pointer values; and
- (e) comparing said input search value and said stored search values in said instances of the search results retrieved in said step (d) to determine whether a respective hash collision has occurred, wherein a presumably usable said instance of the search results is one wherein a hash collision has not occurred, thereby searching the database in a multi-way set-associative manner wherein the size of the database is not a function of the degree of multi-way set-associativity.

[c10] 10. The method of claim 9, wherein said pointer values are equal or smaller in size than said hash address.

[c11] 11. The method of claim 9, wherein the search results each include a stored search value and an associate value, and the method further comprising:

- (f) selecting a usable said associate value, if any, as being that from said usable said instance of the search results.